Early Delirium Detection Using Machine Learning Algorithms

Célia Figueiredo[1(✉)], Ana Cristina Braga[1], and José Mariz[2]

[1] ALGORITMI Centre, University of Minho, 4800-058 Guimarães, Portugal
celianlfg@hotmail.com, acb@dps.uminho.pt
[2] ICVS - Life and Health Sciences Research Institute, 4700-057 Braga, Portugal

Abstract. Delirium is a common manifestation of severe acute neuropsychiatric dysfunction prevalent in hospital settings, which due to the complex multi-factorial causes is often under-diagnosed and neglected. Early detection of delirium is a critical concern that can be effectively addressed using machine learning (ML) techniques. As such, some methods to improve the accuracy of ML classification models for the detection of delirium are covered in this document. The aim of this paper is to develop and validate a tool for use in a hospital setting to accurately identify delirium during the admission of a patient.

A database collected at a Portuguese hospital between 2014 and 2016 was used to conduct this experimental research. Available data comprised 511 records and 124 variables, including patient demographics, medications administered, admission category, urgent admission, hospitalization period, history of alcohol abuse and laboratory results.

The methodologies used included data pre-processing, data imbalance processing, feature selection, train and test model with different ML classifiers, evaluating model performance and development of a Python web-based application.

The model achieved consists of 26 predictors assessed during admission to a healthcare facility. This model combines the *SelectFromModel* method with the logistic regression algorithm, resulting in an area under the receiver operating characteristic curve of 0.833 and an area under the precision-recall curve of 0.582.

Although the prediction model can be enhanced, this approach could be a useful support tool to identify patients at increased risk for delirium in healthcare settings. The application developed is available on: https://bit.ly/3waT3T7.

Keywords: Delirium · Logistic regression · Machine learning · Random forest

This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022 Workshops, LNCS 13377, pp. 555–570, 2022.
https://doi.org/10.1007/978-3-031-10536-4_37

1 Introduction

Over the past few years, the average life expectancy has increased. But along with the ageing population, chronic diseases have increased significantly [9]. These are characterized by high levels of disability and are often responsible for causing pressure on the healthcare system. The biological ageing which an individual is exposed to has a variable course. In this natural degradation, many alterations can occur in the normal functioning of the body. Emphasizing here the deterioration that occurs at the cognitive level, which is a very common impairment in the elderly. A slight impairment in the person caused by a change in the condition of the disease, medication, hospitalization, recent physical deficit, or a combination of these may lead to confused and disoriented states. Delirium is a frequent manifestation of severe acute neuropsychiatric dysfunction, and is widespread in hospital settings. Due to its clinical presentation and variability, it is often under-diagnosed and neglected [21].

Delirium can affect people of all ages, but predominantly affects hospitalized older adults. This geriatric syndrome leads not only to an increase in morbidity and mortality, but also to an increase in the period of hospitalization and a deterioration of the physical and mental state of the individual. The ability to assess delirium is an essential component of the patient assessment strategy for the prevention or treatment of delirium. Some studies suggest that early diagnosis and an appropriate approach are associated with decreased rates of morbidity and mortality associated with delirium [13,21].

In recent years, several clinical assessment tools for delirium have been developed, which represents an important methodological advance in the study and diagnosis of this disorder. Since delirium can easily go unnoticed by healthcare professionals, especially in the intensive care unit (ICU) and emergency departments, it is indispensable to use screening tools that make it possible to detect this disorder faster. Not only to improve the quality of life of patients, but also to reduce patient care costs [23]. Based on this need, screening tools for this disorder have been developed and validated for use in daily clinical practice [8]. Currently, there are several validated instruments to support the diagnosis of delirium that have been adapted according to patients involved [17]. Consequently, there are already more than 30 instruments developed and tested for the assessment of delirium [1,31]. However, in a hospital setting where time is often short and answers must be obtained quickly, it is likely that this type of syndrome goes unnoticed by many medical professionals. Hence, the need has arisen to investigate tools to make diagnosis more quickly and accurately than usual. And, since we already had data collected between 2014 and 2016 in a Portuguese hospital regarding this topic, it was decided to develop a web application that would determine the risk of a patient developing delirium in a hospital setting using ML algorithms.

In this paper, we seek to provide a survey of the current understanding of the delirium problem and to propose a possible solution to aid diagnosis. We will first review the state-of-the-art solutions based on ML to address this problem in Sect. 2. We then describe the data and data techniques used in data

preparation, as well as the ML algorithms used and the metrics for evaluating classifier performance. In Sect. 4 are displayed the experimental results, the best model achieved and the application developed. Finally, a conclusion is provided in Sect. 5.

2 Related Work

The rapid development of artificial intelligence has increasingly allowed the application of ML to fulfil the needs of human life [32]. A study conducted in 2020 by Vellido indicates that the current conjecture of technological development has triggered the idea that the use of ML would be the way forward to solve health-related problems, as well as being an asset to improve the quality of health services [27]. Similarly, another study highlights the potential of ML implementation in healthcare by promoting an improvement in the quality of medicine and allowing to accelerate the pace of evolution of complex diagnostic techniques [14].

The PREdiction of DELIRium for Intensive Care patients (PRE-DELIRIC), was a delirium prediction model created in 2012 for use in adult intensive care patients. The model was developed using data from 1613 consecutive intensive care patients in one hospital and was temporally validated using 549 patients from the same hospital. This model predicts the development of delirium throughout the hospital stay by 10 risk factors (age, Acute Physiology and Chronic Health Evaluation-II (APACHE-II) score, admission group, coma, infection, metabolic acidosis, use of sedatives and morphine, urea concentration, and urgent admission) accessible 24 h after patient admission. The model produced an area under the receiver operating characteristic curve (AUC-ROC) of 0.87 (95% confidence interval: 0.85 to 0.89) and 0.86 after bootstrapping. Temporal and external validation resulted in an AUC-ROC of 0.89 (95% confidence interval: 0.86 to 0.92) and 0.84 (95% confidence interval: 0.82 to 0.87), respectively [26]. According to Liang et al., the PRE-DELIRIC has a high predictive value and the authors suggested that this model can be adopted in ICU to detect delirium in high-risk patients, as it contributes to improving the management of resources, as well as improving patient lives [18].

In 2015, an alternative model for early detection of delirium in intensive care, called Early PREdiction of DELIRium for Intensive Care patient (E-PRE-DELIRIC), was validated. The E-PRE-DELIRIC model uses the data available at ICU admission to predict the development of delirium during the patient's hospital stay. It is composed of nine predictors: age, history of cognitive impairment, history of alcohol abuse, blood urea nitrogen, admission category, urgent admission, mean arterial blood pressure, use of corticosteroids, and respiratory failure. This study emerged as a need to fill the gap in the previous model where predictors are required from the first 24 h after admission to the ICU. Data from 2914 patients was included. The AUC-ROC obtained was 0.76 (95 % confidence interval: 0.73 to 0.77) in the development dataset and 0.75 (95% confidence interval: 0.71 to 0.79) in the validation dataset. An AUC-ROC of 0.70 (95% confidence interval: 0.67 to 0.74) was obtained in the case of delirium that developed until

two days and 0.81 (95% confidence interval: 0.78 to 0.84) for delirium that developed after six days [30]. In order to understand which of the two models would be better prepared for clinical use, the study "Delirium prediction in the intensive care unit: comparison of two delirium prediction models" was conducted in the year 2017. This study concluded that the PRE-DELIRIC model predicts delirium better, however, ICU physicians prefer E-PRE-DELIRIC because it is more convenient to use than PRE-DELIRIC [29].

In 2018, a study was conducted to evaluate the prediction of delirium using the random forest (RF) algorithm. For this purpose, a data collection was previously performed, which involved screening for delirium using Confusion Assessment Method (CAM), and collecting the electronic health data from 64038 patients. The data used included demographic data, comorbidities, medications, procedures, and physiological measures. These data were randomly divided into 80% for training and 20% for testing and applied to the RF algorithm. This predictive model produced an AUC-ROC of 0.909, which demonstrated that this algorithm is highly accurate and has the potential to provide a clinically useful predictive model [7].

More recently, in 2021, a retrospective cohort study was published that developed and validated ML algorithms to detect delirium. To conduct this study, data were collected over 5 years period and the Delirium Observation Screening Scale (DOSS) was applied to inpatients and the Confusion Assessment Method for the Intensive Care Unit (CAM-ICU) for ventilated patients. The information collected included medical history, medications administered, physiological measurements, and laboratory results. The algorithms investigated included RL, Decision Tree, RF, Gradient Boosting Machine, Gaussian Naïve Bayes, Support Vector Machine, and K Nearest Neighbour. This study concluded that RF, gradient-boosted machine, and logistic regression (LR) models demonstrated the best predictive ability with respective AUCs of 0.85 to 0.86 [16].

3 Materials and Methods

3.1 Data

This study used data from a Portuguese hospital that was extracted between 2014 and 2016. The study population comprised patients admitted in the emergency department (ED), patients were aged between 18 and 100 years old. The outcome measure of this study was a positive delirium diagnosis determined by Richmond Agitation Sedation Scale (RASS). The data available included patient demographics, patient origin, admission category, urgent admission, length of stay in the ED, alcohol dependence, date of death, SIRS criteria, medications administered, and lab results. In total, the data provided contained information about 511 individuals admitted to the ED and 124 variables. This data, although a few years old, are still suitable since they correspond to a sample of the ED patient population that has remained constant: many comorbidities and advanced age. This is due to the Portuguese demography which has remained

stable and without major differences due to migratory flows or population fluctuations. Moreover, ED admissions during this period are similar to the current post-pandemic period.

Data Preparation. Consists of raw data analysis techniques to produce quality data, such as data collection, handling missing data, cleaning, transformation and data reduction [33].

ML analysis can frequently be complicated by missing data values. Missing values can occur because values are missing at random or for unknown reasons, and values that, when absent, provide useful information about the task to be performed. Due to the lack of almost 50% of data, the first step was to collect missing data from the hospital's computer system. During collection, it was noted that information on 77 patients was no longer available. As a result, these data set rows were excluded. The next step was to analyse variables, which included the elimination of columns with repeated information and the exclusion of columns with a single value (zero variation). Thus, after collecting missing data, excluding irrelevant variables and deleting rows where information was unavailable, the resulting database comprises information from 434 individuals and 53 independent variables.

The next step was to apply dimensionality reduction, wherein the redundancy of input data was explored. A small set of new variables was created, each being a combination of input variables with essentially the same information. This dimensional reduction consisted of clustering drugs variables and allowed a better organization of the data, as well as a reduction of the dimensionality without loss of relevant information. In this sense, research was conducted to group the drugs by their respective pharmacological group. As a result of this approach, there was a decrease from 53 to 28 independent variables.

The final step involved data transformation. Since ML algorithms at their core operate on numerical data, it was necessary to execute a data transformation to satisfy this requirement. This means that there was a need for categorical variable encoding techniques in this database. In addition, a normalization of the data was also performed because the database consists of 13 numerical variables which comprise different measurement scales and are therefore difficult to compare.

Imbalanced Data. A common roadblock in many ML applications to healthcare is the significant imbalance relation between classes. For example, it is more common to obtain more data on healthy people than on people with an illness. In total, this dataset included 96 positive and 338 negative examples. After splitting data into test and train sets, were obtained 73 positive and 204 negative examples in the training set and the remaining for testing set. Without data balancing, the overall accuracy of the predictions (percentage of predictions that were correct) was 74%. Although this appears reasonable, accuracy is not an appropriate measure to evaluate performance in this context because a null classifier that simply predicts all the samples without delirium achieves nearly the

same accuracy. In this context, it is more appropriate to evaluate separately sensitivity (fraction of delirium detected) and precision (percentage of predicted delirium that is truly delirium).

Ordinarily to address this problem is to apply data balancing techniques to the training data, which can be classified into two major approaches: undersampling and oversampling. The first approach involves reducing the number of examples of the majority class, and the second generates synthetic records of the minority class. In this study, due to the small amount of data, an oversampling technique was considered, namely Adaptive Synthetic Sampling (ADASYN). The main idea behind the ADASYN algorithm is to use a systematic method to create an adaptive way, different amounts of synthetic data according to their distributions [28]. With this approach, new samples of the minority class were synthetically created and a balance of the proportion of the categories was achieved.

In general, the most appropriate performance measure is dependent on the intended application of the classifier. For problems such as identifying health issues, it may be equally important to identify delirium or not. A wide variety of performance measures are used in practice, and in this paper were included the F_1, the AUC-ROC and area under the precision-recall curve (AUC-PR).

Feature Selection - Most of the time, the dataset is made up of numerous variables, some of which may not be relevant or redundant to the classification model. For this reason, prior to constructing the model, it is important to select the best set of independent variables to be included in the predictive model. For this purpose, statistical and exploratory techniques were studied to select and delete variables that contribute less to the model. The main objective is to find a subset of variables that best correlates with the response variable, without eliminating relevant information, allowing a reduction in computational costs and an increase in the predictive power of the classifier. There are three types of algorithms for feature selection: filter methods, wrapper methods, embedded methods [15]. In this project was used a wrapper approach to select features.

In the wrapper approach, the learning algorithm is used to determine the optimal feature subset. Different combinations of subsets are defined, and their performance is evaluated using a classification algorithm. The classification model is first constructed using a subset of samples for training, and then the model is evaluated by the rest of the samples to test the model. The resulting classifier is evaluated according to evaluation metrics such as accuracy, precision, recall, F_1, AUC-ROC, AUC-PR, among others. The subset with the highest outcomes is the subset used to construct the final model. This method is quite suitable for the selection of relevant variables, but implies a high computation cost and can be prone to overfitting [5,25].

Sequential forward selection, sequential backward selection and bidirectional selection are examples of wrapper methods which behave quite similar. All belong to the same group, sequential search, where the method selects only one variable per loop among all successors. The *scikit-learn* [22] presents the *SequentialFeatureSelector* (SFS) function that contains two configurable parameters that allow

to change between methods. The forward strategy begins with no variables in the model, then adds variables to the model one by one. At each step, each remaining variable is tested in order to be included in the model. In each subsequent iteration, the most significant variable is added first. When the newly added variable does not improve model performance, the method ends. In the backward model, on the other hand, the procedure is initiated with all the predictors to be included in the model and in each iteration the least significant variable that allows for an improvement in model performance is eliminated. This process is repeated until no improvement in model performance is observed. The two-way model combines the two techniques mentioned above, and can be combined so that at each step, the procedure selects the best attribute and removes the worst one from among the remaining attributes. It is considered less greedy than the two previous procedures, since it reconsiders the addition of predictors in the model that were removed and vice-versa [6]. The *SelectFromModel* is also a wrapper technique, which extracts the best features of a given dataset according to the importance of weights. The *SelectFromModel* is a meta-estimator that determines the weight importance by comparing to the given threshold value. This function requires an estimator, like RF or LR, which must have attributes to provide the indexes of selected data. Recursive feature elimination (RFE) is another wrapper approach that fits a model and recursively eliminates the least important features based on specific attributes taken by the estimator, until the specified number of features is reached. Finally, the Recursive Feature Elimination with Cross-Validation (RFECV) is an algorithm that eliminates irrelevant features based on validation scores. First, the estimator is trained on the initial set of variables and the importance of each variable is obtained through a specific attribute. Then, the less important variables are pruned from the current set of variables. This procedure is repeated recursively on the pruned set until the desired number of features to be selected is reached [22].

3.2 Machine Learning Algorithms

Random Forest, as its name suggests, is a combination of tree classifiers where each tree depending on a set of random variables, that combines the performance of a wide range of decision tree algorithms to classify or predict the value of a variable [4]. The RF classifier consists of randomly selected features or a combination of features at each node to grow a tree. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. The outcome is classified by taking the most popular voted class from all the tree predictors in the forest [4].

There are several approaches to the selection of attributes used for decision tree induction and most approaches assign a quality measure directly to the attribute. The most frequently used attribute selection measures in decision tree induction are the Information Gain Ratio criterion and the Gini Index [11]. However, *scikit-learn* provides a tool that measures feature importance, also known as the Gini index, by looking at how much the tree nodes that use that

feature reduces impurity across all trees in the forest. The greater the increase in leaf purity, the more important the variable is. The score is automatically computed for each feature after training, and the results are scaled so the sum of all importance is equal to one.

One of the biggest advantages of RF is its versatility. It can be used for both regression and classification tasks [4]. The main limitation of RF is that a large number of trees can make the algorithm too slow and ineffective for real-time predictions.

Logistic Regression is a statistical technique designed to generate observations based on a set of observations, a model that allows the prediction of values taken by a categorical variable, based on one or several continuous and/or binary independent variables. This technique uses similar general principles used in linear regression, the difference is in the response variable which in LR is binary (dichotomous) whereas in the linear regression model it is continuous [12]. The intention behind using LR is to find the best fitting model to describe the relationship between the dependent and the independent variable. LR uses the sigmoid function to map predicted values to probabilities. This function is useful to map any value into another value between 0 and 1. *Scikit-learn* provides a tool that measures the coefficient value of each variable. In general, methods used to find coefficients of a logistic function follow an iterative process of selection of a candidate variable and calculate the logarithm of the probability. This process is repeated until convergence is achieved, and the maximum likelihood is found. Therefore, to interpret the coefficient values we need to calculate the odds ratio (OR).

According to Stoltzfus (2011), regression techniques are versatile when applied to medical research because of their ability to predict outcomes and control variables. He also argues that LR is an effective and powerful way to analyse the impact of a group of independent variables on a binary outcome, quantifying the contribution of each variable [24].

3.3 Metrics for Evaluating Classifier Performance

Classification models should be evaluated before adoption in a real-life setting, because if the classifier is not properly calibrated, it may mislead healthcare professionals and consequently cause harm to the target population. Therefore, in order to minimize such occurrences, it is imperative to evaluate the quality of the resulting forecasts. In order to generate evaluation metrics, the different types of successes or errors must first be represented. This was done using the terms summarized in the confusion matrix: false positive (FP), true negative (TN), true positive (TP) and false negative (FN) [11]. According to previous four outcomes from the confusion matrix, several evaluation measures can be described as follows:

Recall: This may be defined as the probability that the model correctly classifies a person with the delirium syndrome, given that the individual carries the syndrome. This measure is also known as true positive rate (TPR), in other words, this measure assesses the ability of the test to detect delirium when it is indeed present, and its value can be estimated using the following formula [11]:

$$Recall = Sensitivity = TPR = \frac{TP}{TP + FN} \tag{1}$$

Precision: It can be considered as a measure of 'exactness' [11], i.e., it measures the probability of the classifier labelled a person with delirium, given that the person does not have delirium. Its value can be estimated according to the following formula.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$\boldsymbol{F_1}$**:** This metric is the harmonic mean of precision and recall, giving equal weight to both. It evaluates models by considering both FP and FN, and is usually more useful than accuracy if the data have an imbalanced distribution [11].

$$F_1 = \frac{2 * precision * recall}{precision + recall} \tag{3}$$

ROC Curve: One way to evaluate the ability of a diagnostic test to discriminate between two populations is by receiver operating characteristic (ROC) analysis [10]. This analysis is based on signal detection theory and was developed during World War II, where it was used to analyse radar images [3,19]. The science of signal detection theory was later extended to other scientific domains, including diagnostic medicine [20]. Being a commonly used tool in medical diagnostics due to its discriminative capacity [3].

In general, the ROC curve can show the performance of an ML model for binary classification. It is a two-dimensional graphical representation, plotting sensitivity (true positive rate) on the y-axis against 1-specificity (false positive rate) on the x-axis. Whether the test can be affirmed as having the ability to discriminate between individuals with and without delirium is directly linked to a measure of ROC curve accuracy, called the AUC-ROC. Using this measure, it was possible to transform the ROC performance into a scalar value, which allowed to evaluate the discriminant ability of the ROC curve. A closer curve in the upper left corner presents a higher discriminating capacity and the AUC can reach the maximum value of 1, meaning perfect discrimination. On the other hand, if the curve approaches the diagonal, the discriminant capability of the model is null. In summary, the ROC analysis provides important information about diagnostic test performance, and the closer the curve approaches the upper left corner, the greater discriminatory capability of the test.

Precision-Recall Curve: This measure is useful to evaluate the quality classifier's results in cases involving imbalanced datasets, which is an alternative to the ROC curve. The Precision-Recall (PR) curve indicates the balance between precision and recall for different thresholds. There is a great difference between the visual representation of between the ROC curve and the PR curve, because the objective of the ROC space is in the upper left corner while the objective of the PR space is in the upper right corner. Thus, a high AUC-PR represents both high recall and high precision, where high precision is associated with a low false positive rate and high recall is associated with a low false negative rate.

4 Results

This section presents and discusses different models derived from the application of RF (Table 1) and LR (Table 2) classification algorithms and the multiple combinations of feature selection methods. The dataset used was previously transformed by the methods described in Sect. 3.1. The result was a dataset with 343 instances and 38 variables (including features created by one-hot encoding). For testing different algorithms, the dataset was divided: 36% for test and 74% for train.

In Table 1, the results obtained for the feature selection with the RF classification algorithm are shown. Model 0 displays the results obtained without using a feature selection method. This model is used as the control model, i.e., the default model for comparisons between models.

The values selected for the threshold of the *SelectFromModel* method ranged between 0.01 and 0.05 (Model 1 to 5). As the threshold value increases, the number of selected variables decreases. Particularly, a slight variation in the

Table 1. Results obtained by the RF algorithm for feature selection

Random Forest								
	Method	Selected variables	Accuracy	Recall	Precision	F_1	AUC-PR	AUC-ROC
0	–	38	78.34	37.14	52	43.33	0.42	0.789
1	*SelectFromModel* (threshold = 0.01)	37	82.8	45.71	66.67	54.24	0.47	0.808
2	*SelectFromModel* (threshold = 0.015)	17	75.16	31.43	42.31	36.07	0.417	0.765
3	*SelectFromModel* (threshold = 0.02)	14	78.34	31.43	52.38	39.29	0.425	0.796
4	*SelectFromModel* (threshold = 0.03)	13	76.43	34.14	46.43	41.27	0.43	0.76
5	*SelectFromModel* (threshold = 0.05)	9	78.34	42.86	51.72	46.88	0.428	0.748
6	RFECV	31	78.98	40	53.85	45.9	0.508	0.789
7	SFS (forward)	33	78.98	42.86	53.57	45.62	0.445	0.786
8	SFS (backward)	35	80.25	37.14	59.08	45.61	0.457	0.796
9	SFS (both backward)	34	77.71	37.14	50	42.62	0.485	0.785
10	SFS (both forward)	32	77.07	40	48.28	43.75	0.463	0.786

threshold from 0.01 to 0.015 resulted in a decrease of 20 variables. In general, models 1 and 3 show a slight improvement of AUC-ROC in comparison with Model 0. For the remaining thresholds, there was no significant improvement in the AUC-ROC and AUC-PR. An alternative method tested was the RFECV, which corresponding to model 6 (Table 1) was the one that obtained the best results when comparing the AUC-PR. The number of variables decreased from 38 to 31 and the AUC-PR increased by 0.088. Finally, the SFS was tested. The models corresponding to this method (Table 1) are 7, 8, 9 and 10, and only Model 8 showed a slight improvement and the remaining models did not show any significant improvement compared to model 0 when comparing the AUC-ROC. After observing the AUC-PR column, it was found that only model 6 had a value slightly higher than 0.50. These findings suggest that none of the tested models is capable of making quality predictions, since this metric is widely used in the evaluation of forecasting models with imbalanced data.

Table 2. Results of feature selection with LR algorithm (**bold** denotes best model)

Logistic Regression								
Model	Method	Selected variables	Accuracy	Recall	Precision	F_1	AUC-PR	AUC-ROC
0	–	38	83.44	60	63.64	61.76	0.579	0.8320
1	SFS (forward)	23	80.25	48.57	56.67	52.31	0.506	0.766
2	SFS (backward)	26	78.98	57.14	52.63	54.79	0.500	0.802
3	SFS (both forward)	27	70.06	54.529	38	44.71	0.4590	0.755
4	SFS (both backward)	19	78.98	57.14	52.63	54.79	0.5260	0.788
5	RFE	19	80.25	60	55.26	57.53	0.565	0.7827
6	RFECV	18	78.34	51.43	51.43	51.43	0.543	0.755
7	*SelectFromModel* (threshold = 0.02)	38	83.44	60	63.64	61.76	0.579	0.8320
8	***SelectFromModel* (threshold = 0.09)**	**36**	**84.71**	**60**	**67.74**	**63.64**	**0.582**	**0.8333**
9	*SelectFromModel* (threshold = 0.1)	35	84.08	60	65.62	62.69	0.578	0.8311
10	*SelectFromModel* (threshold = 0.2)	33	84.08	60	65.62	62.69	0.575	0.8278
11	*SelectFromModel* (threshold = 0.3)	27	82.17	60	60	60	0.563	0.8290
12	*SelectFromModel* (threshold = 0.5)	24	80.89	60	50.76	58.33	0.542	0.8208
13	*SelectFromModel* (threshold = 1)	14	77.71	40	50	44.44	0.499	0.7250
14	*SelectFromModel* (threshold = 1.5)	7	70.7	54.29	38.78	45.24	0.509	0.7207

Table 2 shows the results achieved by the different feature selection methods with the LR classification method. Similar to the previous classification model,

Model 0 was also registered, which corresponds to the results achieved without the feature selection technique. Models 1 to 4 are related to the SFS algorithm, and none of them improved the evaluation metrics obtained in Model 0.

The models RFE and RFECV reduced the number of variables to 19 and 18 respectively, and the AUC-PR values although slightly lower, are close to AUC-PR value obtained by Model 0. Finally, models 7 to 14 depict results obtained by the *SelectFromModel* function. It is possible to verify that Model 8 performed best by comparing the AUC-ROC and AUC-PR metrics with the other models. Models 7, 9, 10 and 11 also provide reasonable results for this project, with the AUC-PR showing results of almost 0.57 and the AUC-ROC near 0.83. Of these models mentioned, the one that presents the best results in terms of model dimensionality is Model 11, as it selects 27 variables allowing for a significant decrease in model dimensionality. However, Model 8 with 36 variables was chosen, since the best value obtained by the evaluation metrics was prioritized. Models 12 to 14 were dismissed, because they presented AUC-PR and AUC-ROC values below 0.55 and 0.83, respectively.

The variables selected by Model 8 and the respective coefficients are shown in Table 3. The results show that the coefficient values vary between negative and positive scores. This means coefficients with positive values indicate the variable predicts class delirium. On the other hand, coefficients with negative values determine the absence of delirium.

Table 3. Variables selected by Model 8 with respective coefficients and OR

Variable	Coef	OR	Variable	Coef	OR
Age	2.763	15.852	pH	−40.338	0.713
Other medications	1.589	4.901	Urgent admission	−40.547	0.579
Antipsychotics	1.089	2.974	HCO3	−40.665	0.514
Glucose	0.661	1.937	Intra-Hospital	−40.692	0.500
Urea	0.605	1.831	Home	−0.717	0.488
CO2	0.565	1.759	Analgesics	−0.778	0.459
Length of stay	0.560	1.751	Gender	−0.919	0.399
Creatinine	0.356	1.428	Hemato-Oncology	−1.187	0.305
CRP	0.301	1.351	Neurology	−1.268	0.281
Anticoagulants	0.287	1.332	Respiratory	−1.312	0.269
SIRS criteria	0.268	1.307	Nursing home	−1.316	0.268
Alcohol dependence	0.250	1.284	Other admission category	−1.334	0.264
Corticosteroids	0.230	1.258	Inter-Hospital	−1.481	0.227
Antidepressants	0.093	1.098	Cardiovascular	−1.829	0.160
Digitalis	−0.140	0.869	O2	−1.844	0.158
Sodium	−0.184	0.832	Musculoskeletal	−2.006	0.134
Ionized calcium	−0.199	0.819	Genitourinary	−2.079	0.125
Antidyslipidemics	−0.303	0.738	Gastrointestinal	−2.221	0.108

The variables most contributive to the prediction of delirium are age, other medications and antipsychotics. The 'Antipsychotic' variable is composed of the medicines Haloperidol, Quetiapine, Risperidone, Paliperidone and Iloperidone. The variable related to other drugs present includes drugs such as Ranitidine, Scopolamine, Desloratadine, Hydroxyzine, Trihexyphenidyl and Throspium. The inclusion of this variable as one of the most influential in delirium prediction caused some uncertainty, because this group included drugs from several pharmacological groups, which make it impossible to draw any conclusion about which group of drugs has a real effect on the development of delirium. In addition, there are only 18 records in this group, of which 16 (88.89%) are classified as delirium. Based on these statistics, we may consider that this variable may effectively have a significant importance on delirium prediction, since the probability of getting a case of delirium when any of these drugs is present is high. In the case of the variable 'Age', the result shows that for a change of one unit in age, the probability of developing delirium increases 15.8 times. As regards categorical variables, such as the variables 'Other Med' and 'Antipsychotics', the individuals who take this group of drugs are, respectively, 4.9 times and 2.97 times more likely to develop delirium than those who do not take this group of drugs. On the other hand, individuals whose pharmacological therapy contains analgesics or antidyslipidemic drugs are 0.45 times (54%) or 0.73 times (26%) less likely to develop delirium, respectively.

To facilitate the use of this predictive model, a web application was built. This type of format allows easy viewing of the prediction results in real time and on any platform. The purpose of this tool is to easily and quickly calculate the probability of occurrence or absence of delirium in individuals admitted to healthcare settings. The construction of this application implied the use of the prediction model previously selected and Python language. The prediction is performed by filling in all the requested fields in the application form, corresponding to the 26 variables selected by the wrapper method (Fig. 1).

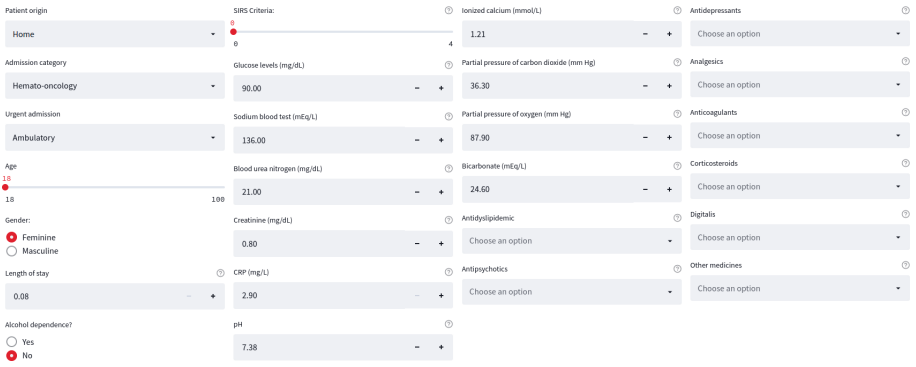

Fig. 1. Screenshot of delirium application developed

The application was developed with *streamlit* and is available through the link: https://bit.ly/3waT3T7. The variables asked on the application form are: patient origin (home, nursing home, intra-hospital, inter-hospital), admission category, urgent admission, age, gender, length of stay, alcohol dependence, SIRS criteria, glucose levels, sodium blood test, blood urea nitrogen, creatinine, C-reactive protein, pH, ionized calcium, partial pressure of carbon dioxide (pCO_2), partial pressure of oxygen (pO_2), bicarbonate (HCO_3), antidyslipidemic (Rosuvastatine, Atorvastatine, Pravastatine, Sinvastatine, Fluvastatine), antipsychotics (Haloperidol, Quetiapine, Risperidone, Paliperidone, Iloperidone), antidepressant (Fluvoxamine, Paroxetine, Sertraline, Venlafaxine, Trazodone, Amitriptyline), analgesics (Nifedipine, Captopril, Clonidine), anticoagulants (Warfarin, Dipyridamole), corticosteroids (Hydrocortisone, Prednisone), digitalis (Digoxin)and other medicines (Ranitidine, Scopolamine, Desloratadine, Hydroxyzine, Trihexyphenidyl, Trospium).

After filling in the application fields, the delirium prediction is calculated. The result is composed of a message generated according to the prediction result.

5 Conclusion

This study demonstrated the importance of data preparation in an ML project to obtain accurate predictions.

The model developed allows the prediction of delirium with an AUC-ROC of 0.833 and an AUC-PR of 0.582. The application developed might be a useful support tool for early delirium detection in healthcare settings. In addition, it is simple to use and adjustable to different types of devices. Although the model obtained has a reasonable predictive capacity, it is postulated that if the set of data collected contained a higher number of records, the predictive model could get a higher hit rate. This assumption is based on the fact that the model was constructed with a reduced number of records and a considerable number of variables. As a consequence, ML algorithms may encounter difficulties to detect patterns. Finally, we conclude that although there is still room for improvement, this predictive model may be an asset in the diagnosis of delirium in hospital settings.

In a future work perspective, another predictive ML model can be built using other classifiers and other feature selection methods. In addition, this project can be improved if the training data set contains more records, namely records related to the development of delirium. Regarding the application, it would be interesting to integrate this type of predictive model into an electronic record of the hospital system and carry out this process automatically in real-time. This approach could reduce the number of cases of delirium that go unnoticed by health professionals, allowing an implementation of a delirium prevention program. Overall, the quality of service could be improved, thus improving the quality of life of hospitalised individuals.

References

1. Adamis, D., Sharma, N., Whelan, P.J.P., MacDonald, A.J.D.: Delirium scales: a review of current evidence. Aging Ment. Health **14**(5), 543–555 (2010). https://doi.org/10.1080/13607860903421011
2. American Psychiatric Association, Diagnostic and statistical manual of mental disorders (5th ed.). Washington DC (2013)
3. Braga, A.C., Oliveira, P.: Diagnostic analysis based on ROC curves: theory and applications in medicine. Int. J. Health Care Qual. Assurance (2003)
4. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
5. Cherrington, M., Thabtah, F., Lu, J., Xu, Q.: Feature selection: filter methods performance challenges. In: 2019 International Conference on Computer and Information Sciences (ICCIS), pp. 1–4. IEEE (2019). https://doi.org/10.1109/ICCISci.2019.8716478
6. Chowdhury, M.Z.I., Turin, T.C.: Variable selection strategies and its importance in clinical prediction modelling. Family Med. Commun. Health **8**(1), e000262 (2020). https://doi.org/10.1136/fmch-2019-000262
7. Corradi, J.P., Thompson, S., Mather, J.F., Waszynski, C.M., Dicks, R.S.: Prediction of incident delirium using a random forest classifier. J. Med. Syst. **42**(12), 1–10 (2018). https://doi.org/10.1007/s10916-018-1109-0
8. De, J., Wand, A.P.F.: Delirium screening: a systematic review of delirium screening tools in hospitalized patients. Gerontologist **55**(6), 1079–1099 (2015). https://doi.org/10.1093/geront/gnv100
9. DuGoff, E.H., Canudas-Romo, V., Buttorff, C., Leff, B., Anderson, G.F.: Multiple chronic conditions and life expectancy. Med. Care **52**(8), 688–694 (2014). https://doi.org/10.1097/MLR.0000000000000166
10. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2006). https://doi.org/10.1016/j.patrec.2005.10.010
11. Han, J., Kamber, M., Pei, J.: Data mining: Concepts and Techniques. Elsevier, Morgan Kaufmann (2012)
12. Hosmer, D.W., Lemeshow, S., Sturdivant, R.X.: Applied Logistic Regression. Wiley Series in Probability and Statistics (2013)
13. Inouye, S.K., Westendorp, R.G.J., Saczynski, J.S.: Delirium in elderly people. Lancet **383**(9920), 911–922 (2014). https://doi.org/10.1016/S0140-6736(13)60688-1
14. Kareemi, H., Vaillancourt, C., Rosenberg, H., Fournier, K., Yadav, K.: Machine learning versus usual care for diagnostic and prognostic prediction in the emergency department: a systematic review. Acad. Emerg. Med. **28**(2), 184–196 (2021). https://doi.org/10.1111/acem.14190
15. Kumar, V.: Feature selection: a literature review. Smart Comput. Rev. **4**(3) (2014). https://doi.org/10.6029/smartcr.2014.03.007
16. Lee, S., Mueller, B., Nick Street, W., Carnahan, R.M.: Machine learning algorithm to predict delirium from emergency department data. BMJ (2021). https://doi.org/10.1101/2021.02.19.21251956
17. Leonard, M.M., et al.: Practical assessment of delirium in palliative care. J. Pain Symptom Manage. **48**(2), 176–190 (2014). https://doi.org/10.1016/j.jpainsymman.2013.10.024
18. Liang, S., Chau, J.P.C., Lo, S.H.S., Bai, L., Yao, L., Choi, K.C.: Validation of PREdiction of DELIRium in ICu patients (PRE-DELIRIC) among patients in intensive care units: a retrospective cohort study. Nursing in Critical Care, (August), 1–7 (2020). https://doi.org/10.1111/nicc.12550

19. Lloyd, M.A., Appel, J.B.: Signal Detection theory and the psychophysics of pain: an introduction and review. Psychosom. Med. **38**(2), 79–94 (1976). https://doi.org/10.1097/00006842-197603000-00002
20. Lusted, L.B.: Decision-making studies in patient management. N. Engl. J. Med. **284**(8), 416–424 (1971). https://doi.org/10.1056/NEJM197102252840805
21. Mittal, V., et al.: Delirium in the elderly: a comprehensive review. Am. J. Alzheimers Dis. Other Demen. **26**(2), 97–109 (2011). https://doi.org/10.1177/1533317510397331
22. Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**(2014), 2825–2830 (2012). http://arxiv.org/abs/1201.0490
23. Shintani, A.K., et al.: Costs associated with delirium in mechanically ventilated patients. Crit. Care Med. **32**(4), 955–962 (2004). https://doi.org/10.1097/01.ccm.0000119429.16055.92
24. Stoltzfus, J.C.: Logistic regression: a brief primer. Acad. Emerg. Med. **18**(10), 1099–1104 (2011). https://doi.org/10.1111/j.1553-2712.2011.01185.x
25. Suppers, A., van Gool, A., Wessels, H.: Integrated chemometrics and statistics to drive successful proteomics biomarker discovery. Proteomes **6**(2), 20 (2018). https://doi.org/10.3390/proteomes6020020
26. Van Den Boogaard, M., et al.: Development and validation of PRE-DELIRIC (PREdiction of DELIRium in ICu patients) delirium prediction model for intensive care patients: Observational multicentre study. BMJ (Online), **344**(7845), 17 (2012). https://doi.org/10.1136/bmj.e420
27. Vellido, A.: The importance of interpretability and visualization in machine learning for applications in medicine and health care. Neural Comput. Appl. **32**(24), 18069–18083 (2019). https://doi.org/10.1007/s00521-019-04051-w
28. Vluymans, S.: Learning from imbalanced data. In: Dealing with Imbalanced and Weakly Labelled Data in Machine Learning using Fuzzy and Rough Set Methods. SCI, vol. 807, pp. 81–110. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-04663-7_4
29. Wassenaar, A., et al.: Delirium prediction in the intensive care unit: comparison of two delirium prediction models. Crit. Care **22**(1), 114 (2018). https://doi.org/10.1186/s13054-018-2037-6
30. Wassenaar, A., et al.: Multinational development and validation of an early prediction model for delirium in ICU patients. Intensive Care Med. **41**(6), 1048–1056 (2015). https://doi.org/10.1007/s00134-015-3777-2
31. Wong, C.L., Holroyd-Leduc, J., Simel, D.L., Straus, S.E.: Does this patient have delirium?: value of bedside instruments. JAMA **304**(7), 779–786 (2010)
32. Xia, H., Wang, C., Yan, L., Dong, X., Wang, Y.: Machine learning based medicine distribution system. In: Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019, vol. 2, pp. 912–915. IEEE (2019). https://doi.org/10.1109/IDAACS.2019.8924236
33. Zhang, S., Zhang, C., Yang, Q.: Data preparation for data mining. Appl. Artif. Intell. **17**(5–6), 375–381 (2003). https://doi.org/10.1080/713827180